

DOCUMENT RESUME

ED 116 602

IR 002 465

AUTHOR Wagner, G. R.; McCants, H. H.
TITLE An Interactive Computer Quizzing System to Support the Proctorial System of Instruction.
INSTITUTION Texas Univ., Austin. Project C-BE.
SPONS AGENCY National Science Foundation, Washington, D.C.
REPORT NO EP-3-9-18-72
PUB DATE 18 Sep 72
NOTE 78p.; For related documents see IR 002 463 and 464

EDRS PRICE MF-\$0.76 HC-\$4.43 Plus Postage
DESCRIPTORS College Curriculum; Computer Assisted Instruction; *Individual Tests; On Line Systems; *Proctoring; *Program Descriptions; Tests; *Test Scoring Machines

IDENTIFIERS Computer Based Quizzing System; Proctorial System of Instruction; Project C BE University of Texas Austin

ABSTRACT

The proctorial system of instruction requires the administration of frequent quizzes that measure the student's readiness to proceed to the next unit of instruction. With the support of Project C-BE, a computer program was written which permits students to take quizzes using on-line conversational commands, or to answer the questions at another location and return to the computer terminal at a later time. The system allows the proctor to accommodate twice as many quiz takers, and it provides him with more release time for tutoring students. Students have been receptive to the use of the system. This document details the features of the system, describes the computer files and it provides sample routines used by the quiz takers. (EMH)

 * Documents acquired by ERIC include many informal unpublished *
 * materials not available from other sources. ERIC makes every effort *
 * to obtain the best copy available. Nevertheless, items of marginal *
 * reproducibility are often encountered and this affects the quality *
 * of the microfiche and hardcopy reproductions ERIC makes available *
 * via the ERIC Document Reproduction Service (EDRS). EDRS is not *
 * responsible for the quality of the original document. Reproductions *
 * supplied by EDRS are the best that can be made from the original. *

ED116602

AN INTERACTIVE COMPUTER QUIZZING SYSTEM
TO SUPPORT THE PROCTORIAL SYSTEM OF INSTRUCTION

by

G. R. Wagner and M. M. McCants
The University of Texas at Austin

EP-3/9/18/72

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.

IR 002 465

*The materials contained herein were supported by PROJECT C-BE under Grant GY-9340, "The Use of Computer-Based Teaching Techniques in Undergraduate Science and Engineering Education," from the National Science Foundation to The University of Texas at Austin, Drs. John J. Allan and J. J. Lagowski, Co-Directors.

2/3/4

AN INTERACTIVE COMPUTER QUIZZING SYSTEM TO SUPPORT THE PROCTORIAL SYSTEM OF INSTRUCTION

G. R. Wagner and M. M. McCants
The University of Texas at Austin

1. Introduction

A recent paper described how Keller's PSI (Proctorial System of Instruction) method is being used for teaching applied statistics (2). One aspect of the PSI method is the unit readiness test. A course is divided into several units with a readiness test at the end of each unit. Administration of the readiness tests is one duty of the proctors and a rule of thumb has been to have one proctor for every ten students. If a student takes an average of two attempts to pass a unit, there are fifteen units, and each test takes an average of thirty minutes to administer and grade, a proctor will administer 300 tests and spends 150 hours doing so during the semester. Obviously, the proctor spends a great deal of time administering tests rather than tutoring. This can also cause scheduling problems with students having to get in queues to take quizzes or have them graded.

This paper describes the CBQS (Computer Based Quizzing System) developed by the authors to increase the effectiveness of the proctors and eliminate queues of waiting students. The system has been developed with built-in flexibility that should satisfy any user needs for a computer quizzing system regardless of whether it is used as a part of a PSI system or some other teaching method.

2. The Quizzing System

The system operates on interactive time-sharing and the student selects one of two routes for quiz taking, namely, answer while on-line using conversational commands; or receive a listing of questions and go elsewhere to think about and answer the questions, return to the terminal at a later time, login again, and enter answers. The following explains the use of both procedures.

A. Off line--all questions printed and answers entered later. After logging into the computer the student calls CBQS and an interactive

dialogue begins.

WHAT IS YOUR NAME
? OBRIEN

O'Brien is now verified as being a student in the course.

YOU ARE ELIGIBLE FOR READINESS TEST 3. IS THIS CORRECT
? YES
DO YOU WANT TO ANSWER THE QUESTIONS LATER
? YES

Five questions are now randomly selected from the question base and printed after which he logs out. When the student is ready to answer the questions he logs in and the dialogue resumes.

WHAT IS YOUR NAME
? OBRIEN
DO YOU WISH TO ANSWER QUESTIONS
? YES

FOR EACH ANSWER, TYPE THE QUESTION NUMBER, A SPACE, AND YOUR ANSWER. QUESTIONS MAY BE ANSWERED IN ANY ORDER. WHEN FINISHED, TYPE DONE.

O'Brien now types in answers and after typing DONE the answers are evaluated and results printed. O'Brien's records are automatically updated as a pass or a recycle.

- B. On-line--student controls test taking method. Dialogue is similar to the above except now the student answers NO to the question: DO YOU WANT TO ANSWER QUESTIONS LATER. He then receives the message: YOU HAVE ELECTED TO CONTROL YOUR QUIZ TAKING SESSION ON-LINE. THE AVAILABLE COMMANDS ARE: ANSWER, NEXT, GRADE, PRINT, STATUS, AND DONE.

After receiving a randomly generated question the student then uses the commands which are defined as follows:

ANSWER means the student is entering an answer to a question.

NEXT causes the next question to be printed either with or without answering the previously generated question.

GRADE indicates the student wants an answer just entered to be graded. If he does not issue the command GRADE after answering a question he may return at a later time and change it by typing the question number and his new answer.

PRINT question number permits a student to return to a previously skipped question.

STATUS informs the student if he has any questions still unanswered.

When the student is finished he types DONE, at which time the test results are printed and the student's record is automatically updated.

This system is totally flexible and a student can take a quiz in whatever non-straightforward procedure which he may desire. It provides all the flexibility that he would have in the usual paper and pencil method.

The student is randomly generated five questions for each quiz on units 1-5 and units 7-13. Unit 6 is a mid-term exam with three questions randomly selected from each of the units 1-5. Unit 14 is part one of a final exam with three questions randomly selected from each of units 1-5 and Unit 15 is part two of a final exam with questions from units 7-13.

3. Dynamic Record Keeping

Record keeping is an internal function of CBQS. When the course staff wants to know the status of any student, they request a report that gives complete information on student achievement to date. Example output is as follows:

COMMANDS: NAME DATA, PRINT NAME, PRINT ALL, DELETE NAME
DATA, CHANGE NAME DATA TO DATA

? PRINT LEE

LEE

PASSED UNIT 6

1 R 1/25

1 P 1/28

2 R 2/ 8

2 P 2/15

3 P 2/24

4 R 3/20

4 R 3/21

4 P 3/22

5 P 4/ 4

6 P 4/ 4

7 R 4/ 5

In this report the 1 R /25 indicates that Lee was recycled on unit 1 on January 25. He then passed unit 1 on January 28. If a complete report of all students is desired, the command PRINT ALL is used.

4. Experience with Application

Some very interesting results have been observed in the authors' experience with CBQS. First, it would appear that the proctor: student ratio can be increased from 1:10 to as high as 1:20 with no decrease in the effectiveness of the proctors. Most importantly, the proctor is able to spend more time tutoring individual students.

Secondly, there is a tremendous amount of time required to prepare good questions. A question cannot be ambiguous, must have clearly-defined answer(s) and most important, must be representative of the material required to be covered. To date, several man-weeks have been spent preparing questions for our course consisting of fourteen units.

A third experience has been the acceptance by the students of CBQS. All indications are that the students like it, and feel comfortable using it. This is in marked contrast to an original non-flexible version. Several students have voluntarily remarked that CBQS provides a better learning mechanism than the old pencil and paper method. The student still has the opportunity to discuss missed questions with the proctor including errors due to phonetic misspelling or misunderstood questions. (CBQS accepts 80 percent spelling accuracy as correct.)

Finally, it has been noted that the recycle rate is higher using CBQS. This has been attributed to the "truthfulness" of the computer. Under the old proctor system, a proctor might be talked into, or out of, a given answer.

5. Summary

CBQS has accomplished what it was intended to do. It has provided a comfortable, adaptive computer quizzing environment for students. Proctors in our PSI course have more time to tutor and it appears that the proctor: student ratio may be increased from 1:10 to 1:20.

The students like the CBQS. They show less anxiety and inhibitions than with previous methods including the old proctor administered pencil and paper test. It is providing a better learning experience and now when a student is recycled, he is prone to try again a few hours later. The students can definitely proceed at their own pace with waiting lines for quiz taking virtually eliminated. Finally, the recycle rate is higher due to the inherent "truthfulness" of the computer.

The programs are written in FORTRAN IV for the Control Data 6600 and are available upon request.

Bibliography

1. Hoberock, L. L., B. V. Koen, C. H. Roth and G. R. Wagner, "Theory of PSI Evaluated for Engineering Education", IEEE Transactions on Education, Vol. E-15, No. 1, pp. 25-29, February.
2. Wagner, G. R. and B. Motazed, "The Proctorial System of Instruction Combined with Computer Pedagogy for Teaching Applied Statistics," The American Statistician, June, 1972.

Computer Based Quizzing System Documentation

The CBQS runs on the University of Texas at Austin Control Data 6400/6600 time-sharing system. The CBQS is designed to be a general purpose interactive system for administering quizzes. However, its usage with students taking a PSI course requires that a student answer all questions correctly to be passed to the next unit. The information stored in the records file is the date of each pass or recycle for each unit. The information stored in the usage file is date and time of each log in or log out with the unit answered and the question numbers of those missed.

The CBQS uses features which are machine-dependent, CDC Fortran dependent, or University of Texas System dependent. For example, the procedure of adding data to the student records file requires that the old file be copied, updated, and replaced. This involves permanent file routines, Fortran input/output routines, and the storage of 10 characters per word for greatest efficiency. These features are described in the first three chapters.

There is a special option for overriding student records after the following question:

YOU ARE ELIGIBLE FOR READINESS TEST 3
IS THIS CORRECT

The student may reply: TEST 5 and obtain that unit's test.

The following algorithms may be of interest:

1. General purpose scanner--Subroutine ISCAN
2. Grading routine--GRADE
3. Alphanumeric comparison program for percent correct--FIND

The question and answers file is maintained as a deck of cards and read into file storage. The records and usage files are created interactively and maintained dynamically.

Documentation Table of Contents

- I. Machine Dependent Features
- II. Control Data Fortran Features
- III. University of Texas System Features
- IV. File Descriptions
 - A. Questions and Answers File
 - B. Records File
 - C. Usage File
- V. TESTER Program and Subroutines
- VI. UT System Routines Used
- VII. UPDATER Program and Subroutines.

Machine Dependent Features

I. Ten Characters per Word

The CDC 6600 stores 10 six-bit characters in a single 60 bit word. The standard Fortran format specification is A10.

II. Recovery of Job Name

The recovery of the job name for security checking purposes is performed with a call to the PP routine GSN through an assembly language routine named RA1.

III. Binary Write to Terminal

The subroutine RDLIN performs a binary write to the terminal of the 12 bit multiplexor representation of the character asterisk before every request for input from the terminal.

Control Data Fortran Features

I. R Format Specification and Data Literal

The Control Data Fortran allows an R format specification. This means that alphanumeric characters will be read into the right-most (low-order) portion of a word with zero fill.

Example,

```
      READ (1,10) I
10  FORMAT (RI)
      IF (I .EQ. IRA) STOP
```

If the 1st column of the record read contains the letter A, it is stored in I so that a comparison with the data literal IRA produces equality. The internal value of the A is 1.

II. L Data Literal

For the file names and numeric passwords an L data literal is used.

Example,

DATA LQ/6LQUESTX/

The name QUESTX is placed in the variable LQ left-justified with with zero fill.

III. PROGRAM Card

The PROGRAM card reserves file buffers and lengths (octal).

IV. Hollerith Literals Compared to Integer Variables

In CDC Fortran, Hollerith literals up to 10 characters long may be compared with either integer or floating point variables, but integer variables are used exclusively in this program.

V. Asterisk as FORMAT Hollerith Enclosing Character

The asterisk is used instead of some kind of quote character for Hollerith data in a FORMAT statement.

Example,

10 FORMAT (*HOLLERITH DATA*)

VI. ENCODE and DECODE Statements

- A. ENCODE (number of characters, FORMAT number, variable name) output list
- B. DECODE (number of characters, FORMAT number, variable name) Output list

VII. Octal Literal

An octal literal is followed by the letter B and may be up to 20 digits in length.

IX. AND Masking Operation

The logical connective .AND. can be used as a masking operator in CDC Fortran.

University of Texas System Features

I. Permanent File System

The University of Texas permanent file system allows a user to save files on the disk unit with periodic transfer of the files to or from his magnetic tape. The two routines provided by the system are:

- 1) READPF. Read the permanent file from the disk and make a copy of it for use by this job.
- 2) SAVEPF. Take the current local file and copy it onto the disk to replace the permanent file by that name.

This system can allow errors to occur by having two jobs reading a permanent file (each obtaining its own copy), modifying the file, and replacing the file. The second replacement destroys the modification performed by the first. The possibility of this error can be reduced by having the shortest possible time elapse for the entire updating process (3 to 5 seconds). This error can be eliminated by allowing only a single job to be active (logged in) at one time.

II. IOBIN "Return" function

The updating process has the following steps:

1. Get a copy of the file (READPF)
2. Copy and modify it to file TMFILE.
3. Throw away the original ("Return" function)
4. Rename the modified copy. (RENAME)
5. Save the modified copy. (SAVEPF)

The University of Texas Input/Output routine IOBIN is called with an argument of 70B to perform the "return" function.

III. RENAME Subroutine

A University of Texas subroutine is called to rename a local file.

IV. File TTY

The terminal is addressed as the file named TTY on the University of Texas System.

File Descriptions

I. Questions and Answers File (QUESTX)

This file contains the questions and answers for each unit according to the following specifications:

1. Each unit has a header card:
 - a) Column 1: letter U
 - b) Columns 2-3: unit number
 - c) Columns 4-5: number of questions in this unit
2. The header card is followed by the first question, its answer, the 2nd question, its answer, etc.
3. The first card of each question:
 - a) Column 1: letter Q
 - b) Columns 2-5: anything (can have question number)
 - c) Columns 6-70: question text
4. Subsequent cards of same question:
 - a) Column 1: blank
 - b) Columns 2-5: anything
 - c) Columns 6-70: question text
5. The first card of each answer:
 - a) Column 1: letter A
 - b) Columns 2-5: anything
 - c) Columns 6-75: answer text
 - d) Column 80: letter C if double-length answer
6. Subsequent (alternative) answers:
 - a) Column 1: blank
 - b) Columns 2-5: anything
 - c) Columns 6-75: answer text
 - d) Column 80: letter C if double-length answer

7. Second card of double-length answer
 - a) Columns 1-5: anything
 - b) Columns 6-75: answer text
8. Each unit is followed by an end-of-unit card:
 - a) Column 1: letter R
9. The last unit is followed by an end of file card:
 - a) Column 1: letter E

Example,

- Q What are the first two letters of the alphabet
- A A, B
- B, A

The student must type in either A, B or B, A to be correct.

Example,

- Q How much is $1 + 1$
- A Range 1.5, 2.5

Any answer greater than or equal to 1.5 and less than or equal to 2.5 would be "correct".

II. Records File (RECRDY)

The records file contains the names and status of all students.

1. The first record contains the number of students:
 - a) Column 1-5: number of students (right-justified)
2. The student name record:
 - a) Column 1: blank
 - b) Columns 2-11: student's name (left-justified)
 - c) Columns 12-16: unit passed (right-justified)
 - d) Columns 17-21: number of P/R entries
 - e) Columns 22-26: number of saved question numbers
 - f) 4 columns each: question numbers

3. If the number of P/R entries is non-zero, those entries follow in the format:
 - a) Columns 1-2: blank
 - b) 10 columns each: P/R entry
maximum 7 per record
4. The 10 character P/R entry has the following format:
 - a) 1-2: unit number
 - b) 3: unused
 - c) 4: P(pass) or R(recycle)
 - d) 5: unused
 - e) 6-7: month number
 - f) 8: slash
 - g) 9-10: day number

III. Usage File (USAGEY)

This file contains a record for each question-printing session (log in) and each question-answering session (log out).

Each record:

- a) Column 1: blank
- b) Columns 2-11: student name
- c) Columns 12-16: unit tested
- d) Columns 17-18: blank
- e) Columns 19-28: date (day, month, year)
- f) Columns 29-30: blank
- g) Columns 31-40: time (hour, minute, second)

Log in record: rest is blank

Log out record:

- h) Columns 41-43: number of errors
- i) 3 columns per number: question number missed.

Main Program - TESTER

This program is the master control program. It calls the LOGIN and LOGOUT subroutines at the appropriate times and calls subroutines SYS1, SYS2, or SYS3 depending on the three different things a student can do with CBQS:

1. Receive a printed copy of his questions. (SYS1)
2. Enter answers to those questions. (SYS2)
3. Receive questions and enter answers interactively. (SYS3)

Definition of Files

1. TTY -- the interactive file corresponding to the terminal.
2. QUESTX -- the list of questions and answers.
3. RECRDY -- the student's current status.
4. USAGEY -- the times of log in or log out.
5. TMFILE -- scratch file.

Definition of Variables

ITTY -- name of TTY file
 LQ -- name of QUESTX file
 LR -- name of RECRDY file
 LU -- name of USAGEY file
 TM -- name of TMFILE file
 IQUNT -- current unit being read from QUESTX file

Subroutines Called

LOGIN -- perform log in operations
 READPF -- read permanent file
 RDALF -- read YES or NO from terminal
 SYS1 -- system 1 (print questions)
 SYS2 -- system 2 (accept answers)
 SYS3 -- system 3 (print and accept)
 LOGOUT -- perform log out operations

```

00002 PROGRAM TESTER(ITTY=77,QUESTX,RECRDY,USAGEY=1000,TMFILE)
00002 INTEGER TM
00002 COMMON /FILES/ ITTY,LQ,LR,LU,TM
00002 COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JRF(2)
00002 COMMON /STATUS/NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
C NEXT = CURRENT UNIT (INCLUDING 6,14,15)
C IUN1 = STARTING UNIT NUMBER (1 FOR 6)
C IUN2 = ENDING UNIT NUMBER (5 FOR 6)
C IMODE = 0 TO PRINT OUT QUESTIONS, 1 TO ACCEPT ANSWERS
C NUMBQ = NUMBER OF QUESTIONS TO BE SELECTED FROM EACH UNIT
C NQSV = TOTAL NUMBER OF QUESTIONS SELECTED
C NQUES = NUMBER OF QUESTIONS AVAILABLE IN THIS UNIT
C NCHOS = NUMBER OF QUESTIONS DESIRED FROM THIS UNIT
000002 COMMON /IQUES/ IQUNT,IQN,ITYPE,IQBFN,IQBUF(7),IQBUF2(7),ITEST
000102 COMMON /STUDNT/ NAME1,NUNIT,INUM,IQSTNM(21),IQAT(50)
000102 COMMON /AUX/ IQ,IQSTMD(5),IUSTAN(7,5),IUNMSV(21)
000102 COMMON /NERR/ NERR,NERRHLT(21)
000102 COMMON /EXPANS/ IPOUTA(7)
000002 DATA ITTY/3,ITTY/
000002 DATA LQ/6LQUESTX/,LR/6LRECRDY/,LU/6LUSAGEY/
000002 DATA TM/6LTMFILE/
000002 CALL LOGIN
000003 CALL READPF(4L2+23,LQ,IFLAG)
000006 IQUNT=0
000007 IF (IMODE .EQ. 1) GO TO 824
000011 IF (IUN1 .NE. IUN2) GO TO 1024
000013 WRITE (ITTY,1023)
000017 1023 FORMAT (/// * DO YOU WANT TO ANSWER THE QUESTIONS LATER*)
000017 CALL RDALF(IYES)
000021 IF (IYES .EQ. 3HYES) GO TO 1024
000023 CALL SYS3
000024 GO TO 600
000025 1024 CALL SYS1
000026 GO TO 600
000027 824 CALL SYS2
000030 600 CALL LOGOUT
000031 END
007100

```


Subroutine SYS1

The purpose of this subroutine is to print the questions on the teletype so that the student can answer them later.

Definitions of Variables

IQ = The internal number of the question to be printed.

The external (actual) number is given by IQSTNM(IQ)

IUNI = The unit number for this set of questions.

Subroutines Called

GETUNT -- position QUESTX file at proper unit

CHOSE -- pick questions randomly

PRNTQ -- print question specified by IQ

20

```

SUBROUTINE SYS1
INTEGER TM
COMMON /FILES/ ITTY,LQ,LR,LU,TM
COMMON /STATUS/NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
COMMON /AUX/ IQ,IQSTMD(5),IQSTAN(7,5),IQNMSV(21)
202 CALL GETUNT
CALL CHOSE
WRITE (ITTY,802) IUN1
802 FORMAT (//* UNIT*,I3/)
IQ=0
228 IQ=IQ+1
CALL PRNTQ
IF (IQ .NE. NUMBQ) GO TO 228
IUN1=IUN1+1
IF (IUN1 .LE. IUN2) GO TO 202
RETURN
END
007400

```

Subroutine SYS2

The purpose of this subroutine is to accept answers to questions previously printed by SYS1. The student enters the question number and his answer on a single line. When all answers have been properly entered, the student enters DONE. (An incorrect entry may be reentered until DONE is entered.) Then the program grades the answers, prints the result, and enters new information into the record file (LOGOUT subroutine).

Definition of Variables

IUNI -- this variable, beginning at IUNI, increasing by 1 until IUN2 is reached, is the unit number of the answers to be entered.

IQSTMD -- array for the status of each question (0 = unanswered, 1 = answered)

IVAL -- question number entered by student

IQSTAN -- array containing the student's answer to each question

IQUNT -- position of QUESTX file (0 = rewind)

IQSTNM -- list of question numbers for each unit (IUNI to IUN2) which was read from RECRDY file. If more than one unit is to be answered, there are three questions per unit. Therefore a move of IQSTNM (I) into IQSTNM (I-3) will move the question numbers for the next unit into the current first three positions.

Subroutines Called

RDLINE -- get a new line from the terminal

ISCAN -- process next piece of information from text buffer

GETUNT -- position QUESTX file

GRADER -- grade question (internal number IQ)

USAGE -- record entry into USAGEY file

SUBROUTINE SYS2

INTEGER TM

COMMON /FILES/ ITTY,LO,LR,LU,LM

COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JRF(2)

COMMON /STATUS/ NEXT,IUN1,IUN2,IMODE,NUM,Q,NOUES,NCHOS,NOSV

COMMON /STUDENT/ NAME1,NUNIT,INUM,IUSTNM(21),IDAT(50)

COMMON /AUX/ IU,IUSTMD(5),IUSTAN(1,5),IUNMSV(21)

COMMON /NEHH/ NEHH,NEHMLT(21)

NEHH=0

824 DO 256 IQ=1,NUMBO

256 IUSTMD(IQ)=0

WRITE (ITTY,428) IUN1

828 FORMAT (// * ENTER ANSWERS FOR UNIT*,13/)

260 CALL ROLINE

CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JRF)

IF (IEND.EQ. 1) GO TO 260

IF ((IEND.EQ. -1) .AND. (JBF.EQ. 4MDUNE)) GO TO 270

IF (IEND.NE. 0) GO TO 262

IVAL=XVAL

DO 261 IQ=1,NUMBO

261 IF (IVAL.EQ. IUSTNM(IQ)) GO TO 265

262 WRITE (ITTY,264)

264 FORMAT (* ERROR IN QUESTION NUMBER. PLEASE ENTER AGAIN.*)

GO TO 260

265 DO 266 I=1,7

266 IUSTAN(I,IQ)=INLIN(I)

IUSTMD(IQ)=1

GO TO 260

270 CALL GETUNT

DO 290 IQ=1,NUMBO

290 CALL GRADER

IF (IUN1.EQ. IUN2) GO TO 831

IUN1=IUN1+1

DO 834 I=4,NOSV

834 IUSTNM(I-3)=IUSTNM(I)

GO TO 826

831 WRITE (ITTY,1050)

1050 FORMAT (// * TURN IN THIS TEST SHEET. IF AT ANY TIME YOU HAVE*/

1 * SUGGESTIONS FOR IMPROVEMENT, PLEASE TELL THE PSI STAFF.*//)

REWIND ITTY

CALL USAGE(1)

RETURN

END

000001
000001
000001
000001
000001
000001
000001
000001
000001
000002
000007
000011
000016
000016
000017
000026
000030
000040
000040
000042
000047
000052
000056
000056
000057
000066
000070
000072
000073
000074
000076
000102
000104
000105
000112
000114
000114
000120
000120
000122
000124
000125
000100

22

Subroutine SYS3

The purpose of the system 3 use of the CBQS is to allow the student to receive questions and supply answers interactively. The following inputs are recognized by SYS3:

1. ANSWER - the answer to the current question has been entered.
2. NEXT -- print the next question.
3. PRINT -- print the question whose number is given.
4. STATUS -- print current status.
5. GRADE -- grade the current question.
6. DONE -- give a final summary of all grades and terminate the session.
7. question number followed by text -- the answer to that question has been given.

Definition of Variables

IQUNT -- position of QUESTX file (0 = rewind)
 IQSTMD -- array of status of each question; 0 - not answered, 1 - answered, 2 - graded
 NERR -- number of errors
 IALL -- all questions flag (0 - not printed, 1 - all printed)
 IQ -- internal question number
 IQSTAN -- array of answers to questions

Subroutines Called

GETUNT -- position file QUESTX
 CHOSE -- pick random question numbers
 PRNTQ -- print current question (IQ)
 RDLINE -- read text from terminal
 ISCAN -- scan text
 GRADER -- grade answer to question (IQ)
 USAGE -- record usage in file USAGEY

```

SUBROUTINE SYS3
COMMON /FILES/ ITTY,LQ,LR,LU,LT
COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JBF(2)
COMMON /STATUS/NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
COMMON /IQUES/ IQUNT,IQN,ITYPE,IQBFN,IQBUF(7),IQBUF2(7),ITEST
COMMON /STUDNT/ NAME1,NUNIT,INUM,IQSTNM(21),IDAT(50)
COMMON /AUX/ IQ,IQSTMD(5),IQSTAN(7,5),IQNMSV(21)
COMMON /NERR/ NERR,NERHLT(21)

IQUNT=0
CALL GETUNT
CALL CHOSE
WRITE (ITTY,2)
2 FORMAT (/// YOU HAVE ELECTED TO CONTROL YOUR QUIZ TAKING SESSION
IN-LINE./// THE AVAILABLE COMMANDS ARE: ANSWER, NEXT, GRADE, PH
2./// STATUS, AND DONE./// YOUR FIRST QUESTION IS:*/)

DO 4 I=1,NUMBQ
4 IQSTMD(I)=0
NERR=0
IALL=0
IQ=1
10 CALL PRNTQ
20 CALL RDLIN
CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
IF (IEND .EQ. 1) GO TO 20
IF (IEND .EQ. 0) GO TO 90
IF (JBF .EQ. 6HANSWER) GO TO 30
IF (JBF .EQ. 4HNEXT) GO TO 40
IF (JBF .EQ. 5HPRINT) GO TO 50
IF (JBF .EQ. 6HSTATUS) GO TO 61
IF (JBF .EQ. 5HGRADE) GO TO 80
IF (JBF .EQ. 4HDONE) GO TO 100
WRITE (ITTY,26)
26 FORMAT (* UNRECOGNIZED COMMAND*)
GO TO 20
30 IF (IQSTMD(IQ) .GT. 1) GO TO 36
DO 32 I=1,7
32 IQSTAN(I,IQ)=INLIN(I)
IQSTMD(IQ)=1
GO TO 20
36 WRITE (ITTY,37)
37 FORMAT (* QUESTION ALREADY GRADED*)
GO TO 20
40 IF (IQ .EQ. NUMBQ) GO TO 60
IQ=IQ+1
GO TO 10
50 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
IF (IEND .NE. 0) GO TO 54
DO 52 I=1,NUMBQ
52 IF (IQSTNM(I) .EQ. XVAL) GO TO 56
54 WRITE (ITTY,55)
55 FORMAT (* ILLEGAL QUESTION NUMBER*)
GO TO 20
56 IQ=I
IQUNT=0
CALL GETUNT
GO TO 10
60 IALL=1
61 DO 62 I=1,NUMBQ

```



```

000143      62 IF (IQSTMD(I) .EQ. 0) WRITE (ITTY,63) IQSTNM(I)
000155      63 FORMAT (* NO ANSWER TO QUESTION#I3)
000155      IF (IALL .EQ. 0) GO TO 20
000156      WRITE (ITTY,66)
000161      66 FORMAT (* TO ENTER OR CHANGE ANSWERS, TYPE THE QUESTION NUMBER AND
1/* YOUR ANSWER. WHEN FINISHED, TYPE DONE.*/)
000161      GO TO 20
000162      80 IF (IQSTMD(IQ) .EQ. 0) GO TO 82
000164      IF (IQSTMD(IQ) .EQ. 2) GO TO 36
000166      CALL GRADER
000167      IQSTMD(IQ)=2
000171      GO TO 40
000172      82 WRITE (ITTY,63) IQSTNM(IQ)
000200      GO TO 20
000201      90 IVAL=XVAL
000203      DO 92 J=1,NUMBQ
000210      92 IF (IQSTNM(J) .EQ. IVAL) GO TO 94
000213      WRITE (ITTY,93)
000216      93 FORMAT (* ERROR IN QUESTION NUMBER, PLEASE ENTER AGAIN*)
000216      GO TO 20
000217      94 IF (IQSTMD(J) .GT. 1) GO TO 36
000223      DO 96 I=1,7
000231      96 IQSTAN(I,J)=INLIN(I)
000233      IQSTMD(J)=1
000235      GO TO 20
000236      100 NERR=0
000237      IQUNT=0
000240      CALL GETUNT
000241      DO 290 IQ=1,NUMBQ
000243      290 CALL GRADER
000247      WRITE (ITTY,1050)
000252      1050 FORMAT (/* TURN IN THIS TEST SHEET. IF AT ANY TIME YOU HAVE*/
1 * SUGGESTIONS FOR IMPROVEMENT, PLEASE TELL THE PSI STAFF.*/)
000252      REWIND ITTY
000254      CALL USAGE(1)
000256      IMODE=1
000257      RETURN
000260      END
000100

```


Subroutine PKUNIT

The current course is structured as follows: after units 1 through 5, unit 6 is a review of all of them. After unit 13, unit 14 is a review of units 1 to 5, unit 15 is a review of units 7 to 13. This routine sets that strategy.

Definition of Variables

NCHOS -- number of questions to be chosen from a unit

IUN1 -- number of unit to start this quiz

IUN2 -- number of last unit to be tested during this session

```
SUBROUTINE PKUNIT
COMMON /STATUS/NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
000001 NCHOS=5
000001 IUN1=NEXT
000002 IUN2=NEXT
000003 IF (NEXT .EQ. 6) GO TO 800
000004 IF (NEXT .EQ. 14) GO TO 800
000006 IF (NEXT .EQ. 15) GO TO 804
000010 RETURN
000011 800 NCHOS=3
000012 IUN1=1
000013 IUN2=5
000014 RETURN
000015 804 NCHOS=3
000016 IUN1=7
000017 IUN2=13
000020 RETURN
000021 END
000022
007500
```


Subroutine CHOSE

This routine randomly selects the questions to be used for this unit.

Definition of Variables

IAN5 -- array of chosen question numbers. (0 -- not chosen,
1 -- chosen)

NUMBQ -- number of questions to be chosen

IX -- random integer between 1 and NQUES

IQ -- Internal question number

IQSTNM -- external question number

NQSV -- total number of questions asked this session for
all units.

IQNMSV -- external question numbers for all questions

Function Called

RANF -- get random number between 0 and 1.


```

SUBROUTINE CHOSE
DIMENSION IANS(30)
COMMON /STATUS/NEXT,IUN1,IUN2,IMOVE,NUMBQ,NQUES,NCHOS,NQSV
COMMON /STUDENT/ NAME1,NUNIT,INUM,IQSTNM(21),IDAT(50)
COMMON /AUX/ IQ,IQSTMD(5),IUSTAN(7,5),IQNMSV(21)
000001 201 DO 200 I=1,NQUES
000001 200 IANS(I)=0
000006 NUMBQ=MIN0(NQUES,NCHOS)
000010 DO 220 I=1,NUMBQ
000013 210 IX=RANF(0)*NQUES+1.0
000015 IF (IANS(IX) .NE. 0) GO TO 210
000022 220 IANS(IX)=1
000024 IQ=0
000030 DO 224 I=1,NQUES
000031 IF (IANS(I) .EQ. 0) GO TO 224
000040 IQ=IQ+1
000041 IQSTNM(IQ)=I
000042 NQSV=NQSV+1
000042 IQNMSV(NQSV)=I
000043 224 CONTINUE
000044 RETURN
000047 END
000047
007300

```

Subroutine GETUNT

This routine positions the question file to the proper unit (variable IUN1).

Definition of Variables

ITYPE -- code key in 1st column of QUESTX file records

IQUNT -- unit number

NQUES -- number of questions in this unit

IQN -- current position of file, question number

31

```

SUBROUTINE GETUNT
COMMON /FILES/ ITTY,LQ,LR,LU,LT
COMMON /STATUS/NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
COMMON /IQUES/ IQUNT,IQN,ITYPE,IQBFN,IQBUF(7),IQBUF2(7),ITEST
IF (IQUNT .NE. 0) GO TO 202
REWIND LQ
202 READ (LQ,916) ITYPE,IQUNT,NQUES
916 FORMAT (R1,2I2)
IF (ITYPE .EQ. 1RE) STOP
IF (ITYPE .NE. 1RU) GO TO 202
IF (IQUNT .NE. IUN1) GO TO 202
IQN=0
RETURN
END

```


Subroutine PRNTQ

This routine prints a question from the current unit. The question number is IQSTNM (IQ).

Definition of Variables

IQBUF -- question of answer buffer

ITEST -- flag (C means answer is two cards)

IQN -- question number, file position counter

SUBROUTINE PRNTQ

```

000001 COMMON /FILES/ ITTY,LQ,LR,LU,LT
000001 COMMON /IQUES/ IQUNT,IQN,ITYPE,IQBFN,IQBUF(7),IQBUF2(7),ITEST
000001 COMMON /STUDNT/ NAME1,NUNIT,INUM,IQSINM(21),IDAT(50)
000001 COMMON /AUX/ IQ
000001 230 IF (IQN.EQ. IQSTNM(IQ)) GO TO 245
000004 240 READ (LQ,918) ITYPE,IQBUF
000014 918 FORMAT (H1,4X,7A10,4X,H1)
000014 IF (ITYPE.NE. 1RQ) GO TO 240
000016 IQN=IQN+1
000020 GO TO 230
000020 245 WRITE (ITTY,1020) IQN,IQBUF
000030 1020 FORMAT (I//,2,1H,,2A,6A10,A5)
000030 250 READ (LQ,918) ITYPE,IQBUF,ITEST
000042 IF (ITYPE.NE. 1R ) RETURN
000045 WRITE (ITTY,1022) IQBUF
000053 1022 FORMAT (5X,6A10,A5)
000053 GO TO 250
000054 END
007300

```


Subroutine GETANS

GETANS gets the answer for the current question so that the GRADE routine can compare it to the student's answer.

Input Parameter

ICODE -- 0 means the routine is looking for the first answer to the current question. This answer has an A in column one. ICODE of 1 means the routine is to look for alternative answers after the first.

Definition of Variables

ITYPE -- column 1 character of current buffer
IQBUF -- current question buffer
ITEST -- code indicating question continues on the next card
IQN -- number of current question being read from QUESTX file
IPOUTA -- correct answer to be printed if student is incorrect
IQBFN -- number of characters in IQBUF buffer

Output Parameter

IFND -- 0 means another answer to the question was not found.
1 means an answer has been found and is in the IQBUF buffer.

```

SUBROUTINE GETANS(ICODE,IFND)
COMMON /FILES/ ITTY,LQ,LR,LU,LT
COMMON /IQUES/ IQUNT,IQN,ITYPE,IQBFN,IQBUF(7),IQBUF2(7),ITEST
COMMON /STUONT/ NAME1,NUNIT,INUM,IQSTNM(21),IDAT(50)
COMMON /AUX/ IQ,IQSTMD(5),IQSTAN(7,5),IQNHSV(21)
COMMON /EXPANS/ IPOUTA(7)
IF (ICODE .NE. 0) GO TO 500
200 IF (IQN .EQ. IQSTNM(IQ)) GO TO 294
210 READ (LQ,918) ITYPE,IQBUF,ITEST
918 FORMAT (H1,4X,7A10,4X,H1)
20024 IF (ITYPE .NE. 1RQ) GO TO 210
IQN=IQN+1
GO TO 200
292 READ (LQ,918) ITYPE,IQBUF,ITEST
294 IF (ITYPE .NE. 1RA) GO TO 292
DO 296 I=1,7
296 IPOUTA(I)=IQBUF(I)
300 IQBFN=70
IF (ITEST .NE. 1RC) GO TO 401
READ (LQ,924) IQBUF2
924 FORMAT (5X,7A10)
IQBFN=140
401 IFND=1
RETURN
500 READ (LQ,918) ITYPE,IQBUF,ITEST
IF (ITYPE .EQ. 1R ) GO TO 300
IF (ITYPE .EQ. 1RQ) IQN=IQN+1
IFND=0
RETURN
END

```


Subroutine GRADER

This routine grades the student's answer and prints one of the three possible results: correct, incorrect, or no answer was given.

Definition of Variables

IFND -- 0 means GETANS did not find another answer, 1 means an answer is available

IPT -- scanner pointer for IQSTAN buffer

IBFN -- number of characters in IQSTAN buffer

NERR -- number of errors

NERRLT -- list of question numbers missed

IGRADE -- 1 means answer is correct

Subroutines Called

GETANS -- get next answer for question IQ

ISCAN -- scanner

GRADE -- compare answer with student's answer

```

SUBROUTINE GRADER
COMMON /FILES/ ITTY,LQ,LR,LU,LT
COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JBF(2)
COMMON /STUDENT/ NAME1,NUNIT,INUM,IQSTNM(21),IDAT(50)
COMMON /AUX/ IQ,IQSTMD(5),IQSTAN(7*5),IQNMSV(21)
COMMON /NERR/ NERR,NERRLT(21)
COMMON /EXPANS/ IPOUTA(7)
CALL GETANS(0,IFND)
IF (IQSTMD(IQ) .EQ. 0) GO TO 294
270 IPT=1
    IBFN=70
    CALL ISCAN(IQSTAN(1,IQ),IBFN,IPT,XVAL,IEND,JBFN,JBF)
    CALL GRADE(IGRADE)
    IF (IGRADE .NE. 0) GO TO 284
    CALL GETANS(1,IFND)
    IF (IFND .NE. 0) GO TO 270
    WRITE (ITTY,1026) IQSTNM(IQ),IPOUTA
1026 FORMAT (13,1H,*,* EXPECTED ANSWER WAS *7A10)
    GO TO 282
294 WRITE (ITTY,295) IQSTNM(IQ)
295 FORMAT (* NO ANSWER TO QUESTION*13)
282 NERR=NERR+1
    NERRLT(NERR)=IQSTNM(IQ)
    RETURN
284 WRITE (ITTY,1028) IQSTNM(IQ)
1028 FORMAT (13,1H,*,* CORRECT*)
    RETURN
END
007100

```


Subroutine GRADE

This routine compares a single correct answer against the student's answer and gives its verdict: correct or incorrect. The following cases are handled:

1. Alphabetic and non-numeric characters. The student must have at least 80% of the characters correct.
2. Exact numerical values. The student's answer must agree exactly with the expected answer.
3. Range of numerical values. The student's answer must be within the acceptable range.

Multiple answers may be required by separating them with commas on the expected answer card. The student must give correct replies for all answers to be graded correct.

If alternative correct answers are to be processed, the GRADER routine determines this and gives a new call to GRADE.

Output Parameter

IGRADE -- 1 means correct, 0 means incorrect

Definition of Variables

ITAB -- table for FIND routine

IQPT -- buffer pointer for IQBUF buffer

Subroutine Called

ISCAN -- scanner

FIND -- alphabetic compare routine

```

SUBROUTINE GRADE(IGRADE)
DIMENSION ITAB(3)
DIMENSION JQBF(2)
COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JRF(2)
COMMON /IQUES/ IQUNT,IUN,ITYPE,IQBFN,IQBUF(7),IQBUF2(7),ITEST
COMMON /AUX/ IQ,IQSTMD(5),IQSTAN(7,5),IQNMSV(21)
IQPT=1
400 CALL ISCAN(IQBUF,IQBFN,IQPT,XQVAL,IQEND,JQBFN,JQBF)
IF (IQEND.EQ. 1) GO TO 1027
CALL ISCAN(IQSTAN(1,IQ),IBFN,IPT,XVAL,IEND,JBFN,JBF)
IF (IQEND.EQ. 0) GO TO 410
IF (JQBF(1).EQ. SHRANGE) GO TO 415
ITAB(1)=JQBFN
ITAB(2)=JQBF(1)
ITAB(3)=JQBF(2)
CALL FIND(JBFN,JBF,ITAB,IVAL)
IF (IVAL.GE. 80) GO TO 400
GO TO 280
410 IF (XQVAL.EQ. XVAL) GO TO 400
GO TO 280
415 CALL ISCAN(IQBUF,IQBFN,IQPT,XQVAL,IQEND,JQBFN,JQBF)
IF (XVAL.LT. XQVAL) GO TO 280
CALL ISCAN(IQBUF,IQBFN,IQPT,XQVAL,IQEND,JQBFN,JQBF)
IF (XVAL.GT. XQVAL) GO TO 280
GO TO 400
1027 IGRADE=1
RETURN
280 IGRADE=0
RETURN
END
000002
000002
000002
000002
000002
000002
000003
000012
000015
000025
000027
000031
000032
000033
000035
000040
000043
000044
000046
000047
000056
000062
000070
000075
000075
000076
000077
000100
000100
007200

```


Subroutine LOGIN

This routine performs log in operations and one mode of operation test.

- I. The job name is checked for validity.
- II. The random number generator is initialized.
- III. The students name is looked up in the RECRDY file.
- IV. A. If questions have been previously printed, the routine asks "Do you wish to enter answers."
B. If the answer to A is Yes, PKUNIT is called and one of two explanations is printed.
- V. If questions are to be printed, the correct test unit number is verified, the log in update of the USAGEY file is performed, and PKUNIT is called.

Definition of Variables

IT -- number of random numbers to be discarded before the next one is used.

NAME1 -- name in RECRDY file

NUNIT -- last unit passed

INUM -- number of pass/recycle entries

NUMBQ -- number of questions printed at last log in to be answered

IQSTNM -- question numbers saved, (to be answered).

NEXT -- unit to be tested on

NQSU -- number of questions printed and saved to be answered later (initialized)

IMODE -- flag, 0-print questions, 1-accept answers

NUMBQ -- number of questions to be answered from each unit

Subroutines Called

RA1 -- get job name

READPF -- get permanent file

SECOND -- get current time used
RANF -- get random number
RDALF -- get word from terminal
IOBIN -- input/output routine
ISCAN -- scanner
USAGE -- add a record to USAGEY file
PKUNIT -- get NCHOS, IUN1, IUN2


```

SUBROUTINE LOGIN
INTEGER TM
COMMON /FILES/ ITTY,LQ,LR,LU,TM
COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JRF(2)
COMMON /STATUS/NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
COMMON /STUDNT/ NAME1,NUNIT,INUM,IUSTNM(21),IDAT(50)
CALL RAL(4LGSNP,JOBNAM)
IF ((JOBNAM .AND. 3777777777700000000000) .EQ.
1 1505031436000000000000) GO TO 30
WRITE (ITTY,20)
20 FORMAT (* ACCOUNT NUMBER MECL360 MUST BE USED*)
STOP
30 CALL READPF(4L2423,LR,IFLAG)
CALL SECOND(TIME)
IT=(TIME-IFIX(TIME))*1000.
DO 10 I=1,IT
10 X=RANF(0)
WRITE (ITTY,1002)
1002 FORMAT (// * PROGRAM TESTER*/// * WHAT IS YOUR NAME*)
90 CALL RDALF(INAME)
REWIND LR
READ (LR,902) NSTONT
902 FORMAT (15)
IF (NSTONT .EQ. 0) STOP
DO 100 ICOPY=1,NSTONT
READ (LR,904) NAME1,NUNIT,INUM,NUMBQ,(IUSTNM(I),I=1,NUMBQ)
904 FORMAT (1X,A10,3I5,21I4)
IF (NAME1 .EQ. INAME) GO TO 120
IF (INUM .EQ. 0) GO TO 100
READ (LR,906) (IDAT(ID),ID=1,INUM)
906 FORMAT (2X,7A10)
100 CONTINUE
WRITE (ITTY,1004)
1004 FORMAT (// * NAME NOT RECOGNIZED*/* PLEASE TRY AGAIN*)
GO TO 90
120 CALL IOBIN(70B,LR)
NEXT=NUNIT+1
IF (NUMBQ .NE. 0) GO TO 258
122 WRITE (ITTY,1006) NEXT
1006 FORMAT (// * YOU ARE ELIGIBLE FOR READINESS TEST*13./
1 * IS THIS CORRECT*/)
CALL RDALF(IYES)
NQSV=0
IF (IYES .EQ. 3HYES) GO TO 124
IF (IYES .NE. 4HTEST) STOP
CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
NEXT=XVAL
IF (NEXT .GT. 15) STOP
124 CALL USAGE(0)
CALL PKUNIT
IMODE=0
RETURN
258 WRITE (ITTY,259)
259 FORMAT (* DO YOU WISH TO ENTER ANSWERS*)
CALL RDALF(IYES)
IF (IYES .EQ. 2HNO) GO TO 122
IF (IYES .NE. 3HYES) GO TO 258
CALL PKUNIT

```

```
000176      IMODE=1
000177      IF (IUN1.NE. IUN2) GO TO 820
000201      WRITE (ITTY,1024)
000205      1024 FORMAT (// * FOR EACH ANSWER, TYPE THE QUESTION NUMBER, A SPACE, *
      .      1 /* AND YOUR ANSWER. QUESTIONS MAY BE ANSWERED IN ANY ORDER.*
      .      2 /* WHEN THROUGH, TYPE DONE.*//)
000205      RETURN
000206      820 WRITE (ITTY,822)
000212      822 FORMAT (// * FOR EACH UNIT, TYPE THE ANSWERS TO ITS QUESTIONS AND
      .      1 ONE.*/* FOR EACH ANSWER, TYPE THE ANSWER NUMBER AND YOUR ANSWER*/*
000212      NQSV=NUMBQ
000213      NUMBQ=3
000215      RETURN
000215      END
006300
```


Subroutine LOGOUT

This routine performs log out operations by updating the RECRDY file to reflect the results of the session:

1. Questions were printed only.
2. Questions were answered, student was passed or recycled.

Definition of Variables

IMNT -- 3 character names of months to convert to an integer
 NSTDNT -- number of students
 NAME2 -- name of student
 NUNIT2 -- unit passed
 INUM2 -- number of pass/recycle entries
 NUMBQ2 -- number of IQSTN2 entries
 IQSTN2 -- numbers of questions to be answered
 IDAT -- pass/recycle information
 IDATE -- system date
 IDY -- day
 IMNTH -- month name
 IMN -- month number
 IPR -- P or R for pass or recycle
 NUNIT2 -- updated last unit passed

Subroutines Called

READPF -- read permanent file
 TIM -- get date
 IOBIN -- input/output routine
 RENAME -- change file name
 SAVEPF -- store permanent file

```

SUBROUTINE LOGOUT
INTEGER TM
DIMENSION IQSTN2(21)
DIMENSION IMNT(12)
COMMON /FILES/ ITTY,LQ,LR,LU,TM
COMMON /STATUS/NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
COMMON /STUDENT/ NAME1,NUNIT,INUM,IQSTNM(21),IDAT(50)
COMMON /AUX/ IQ,IQSTMD(5),IQSTAN(7,5),IQNMSV(21)
COMMON /NERR/ NERR,NERHLT(21)
DATA IMNT/3HJAN,3HFEB,3HMAR,3HAPR,3HMAY,3HJUN,3HJUL,3HAUG,3HSEP,
1 3HOCT,3HNOV,3HDEC/
CALL READPF (4L2423,LR,IFLAG)
REWIND LR
REWIND TM
READ (LR,902) NSTONT
902 FORMAT (15)
WRITE (TM,902) NSTONT
DO 640 ICOPY=1,NSTONT
READ (LR,904) NAME2,NUNIT2,INUM2,NUMBQ2,(IQSTN2(I),I=1,NUMBQ2)
904 FORMAT (1X,A10,3I5,2I4)
IF (INUM2 .NE. 0) READ (LR,906) (IDAT(ID),ID=1,INUM2)
906 FORMAT (2X,7A10)
IF (NAME2 .NE. NAME1) GO TO 620
IF (IMODE .EQ. 0) GO TO 610
CALL TIM(1,IDATE)
DECODE (10,602,IDATE) IDY,IMNTH
602 FORMAT (1X,I2,1X,A3,3X)
IMN=0
DO 604 I=1,12 → N
604 IF (IMNTH .EQ. IMNT(I)) IMN=I
IPR=1HR
IF (NERR .GT. 0) GO TO 606
IPR=1HP
IF (NUNIT2 .LT. NEXT) NUNIT2=NEXT
606 ENCODE (10,608,IDW) NEXT,IPR,IMN,IDY
608 FORMAT (I2,1X,A2,I2,1H/,I2)
INUM2=INUM2+1
IDAT(INUM2)=IDW
WRITE (TM,904) NAME2,NUNIT2,INUM2
GO TO 630
610 NEXT=NEXT-1
WRITE (TM,904) NAME2,NEXT,INUM2,NQSV,(IQNMSV(I),I=1,NQSV)
GO TO 630
620 WRITE (TM,904) NAME2,NUNIT2,INUM2,NUMBQ2,(IQSTN2(I),I=1,NUMBQ2)
630 IF (INUM2 .NE. 0) WRITE (TM,906) (IDAT(ID),ID=1,INUM2)
640 CONTINUE
REWIND LR
REWIND TM
CALL IOBIN(70B,LR)
CALL RENAME (TM,LR,IFLAG)
CALL SAVEPF (4L2423,LR,4L3068,IFLAG)
CALL IOBIN(70B,LR)
RETURN
END

```


Subroutine USAGE

This routine updates the usage log on the USAGEY file. If there were errors in the student's answers, the questions missed are recorded. The maximum length of the file is set at 50 entries.

Definition of Variables

NUS -- number of entries in USAGE file

NUSP1 -- updated number of entries

INBUF -- buffer for file copy operation

Subroutines Called

READPF -- read permanent file

TIM -- get date, time

IOBIN -- input/output routine

RENAME -- change name of a file

SAVEPF -- store permanent file

```

SUBROUTINE USAGE (LOG)
INTEGER TM
DIMENSION INBUF(8)
COMMON /FILES/ ITTY,LQ,LR,LU,LM
COMMON /STATUS/ NEXT,IUN1,IUN2,IMODE,NUMBQ,NQUES,NCHOS,NQSV
COMMON /STUDNT/ NAME1,NUNIT,INUM,IQSTNM(21),IDAT(50)
COMMON /NERR/ NERR,NERRLT(21)
CALL READPF (4L2423,LU,IFLAG)
REWIND LU
REWIND TM
910 READ (LU,910) NUS
    FORMAT (I5)
    NUSP1=NUS+1
    IF (NUSP1.GT. 50) NUSP1=50
    WRITE (TM,910) NUSP1
    CALL TIM(1,IDATE)
    CALL TIM(2,IHRMN)
    IF (LOG.NE. 0) GO TO 420
    WRITE (TM,1042) NAME1,NEXT,IDATE,IHRMN
    GO TO 430
420 IF (NERR.GT. 0) GO TO 425
    WRITE (TM,1042) NAME1,NEXT,IDATE,IHRMN,NERR
    GO TO 430
425 WRITE (TM,1042) NAME1,NEXT,IDATE,IHRMN,NERR,(NERRLT(I),I=1,NERR)
1042 FORMAT (1X,A10,15,2X,A10,2X,A10,2,13)
430 IF (NUS.EQ. 0) GO TO 140
    DO 440 I=1,NUS
    READ (LU,912) INBUF
912 FORMAT (8A10)
440 WRITE (TM,912) INBUF
140 REWIND TM
REWIND LU
CALL IOBIN(70B,LU)
CALL RENAME(TM,LU,IFLAG)
CALL SAVEPF (4L2423,LU,4L3068,IFLAG)
CALL IOBIN(70B,LU)
RETURN
END
000002
000002
000002
000002
000002
000002
000005
000010
000012
000020
000021
000023
000026
000034
000036
000040
000042
000055
000057
000062
000077
000101
000122
000123
000124
000126
000133
000134
000146
000150
000152
000154
000157
000162
000164
000165
000700

```


Subroutine RA1 —

This routine places a PP request at RA+1 and waits for a reply. The only use of this routine is to obtain the job name for LOGIN as a security check.

RA1

COMPASS - V

0	00000000000000000000	RA1	IDENT	RA1
1	56110		ENTRY	RA1
	76220		DATA	0
	12612		SA1	B1
2	5150000001	WAIT1	SX2	B2
	0315000002 +		BX6	X1+X2
3	5160000001		SAS	1
4	5150000001	WAIT2	NZ	X5+WAIT1
	0315000004 +		SA6	1
5	0400000000 +		SA5	1
6			NZ	X5+WAIT2
			EQ	RA1
			END	

46054

STORAGE USED
6600 ASSEMBLY

13 STATEMENTS
0.105 SECONDS

Subroutine RDLIN

This routine reads a new line from the terminal. An asterisk is printed first.

Definition of Variables

ISTAR -- binary representation of an asterisk

INLIN -- terminal input buffer

IPT -- buffer pointer

IBFN -- length of buffer

Subroutine RDALF

This routine obtains the first word typed in by the student. It is called to obtain the student's name and for yes or no answers.

Output Parameter

NAME -- the word

Subroutines Called

RDLINE -- read a line from the terminal

ISCAN -- scanner.

```
000002 SUBROUTINE RDALF(NAME)  
000002 COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JBF(2)  
000003 CALL RDLIN  
000012 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)  
000014 NAME=JBF  
000015 RETURN  
007400 END
```


Subroutine ISCAN

This is the free form scanner routine.

Input Parameters

INLIN -- buffer of text

N -- buffer length

I -- buffer pointer

Output Parameters

X -- numeric value

IEND -- 0 means number found, -1 means non-numeric found,
1 means end of line found

IBFN -- number of characters in JBF buffer

IBF -- buffer containing non-numeric characters

Definition of Variables

IF0 -- letter 0 (zero)

IF9 -- letter 9

IPNT -- letter . (period)

BLANK -- letter (blank)

MINUS -- letter -

PLUS -- letter +

ICA -- letter A

ICZ -- letter Z

ICM -- letter ,

IFLAG -- current status

X -- numeric answer

XS -- numeric sign

XF -- fractional multiplier

Subroutines Called

PULL -- get a character

PUSH -- store a character


```

SUBROUTINE ISCAN(INLIN,N,I,X,IEND,JBFN,JBF)
INTEGER BLANK
DIMENSION INLIN(8),JBF(2)
DATA IF0,IF9,IPNT,BLANK,MINUS,IPLUS,IR0,IR9,IR,,IR ,IR-,IR+,/
DATA ICA,ICZ,ICM,IRA,IAZ,IR,/
IFLAG=0
X=0.0
XS=1.0
XF=.1
JBF(1)=1H
IF(I-N)1,1,39
1 CALL PULL(I,INLIN,K)
IF(K-IF0)+,70,9898
9898 IF(IF9-K)+,70,70
4 IF(K-IPNT)5,60,5
5 IF(IFLAG-2)10,+0,+0
10 IF(K-BLANK)15,100,15
15 IF(IFLAG-1)20,+0,+0
20 IF(K-MINUS)25,50,25
25 IF (K-IPLUS) 30, 51, 30
30 JBF(2)=1H
JBFN=1
CALL PUSH(JBFN,JBF,K)
35 I=I+1
IF (I .GT. N) GO TO 38
CALL PULL(I,INLIN,K)
IF (K .EQ. BLANK) GO TO 36
IF (K .EQ. ICM) GO TO 38
IF (JBFN .EQ. 20) GO TO 35
JBFN=JBFN+1
CALL PUSH(JBFN,JBF,K)
GO TO 35
36 IF (JBF .EQ. 6HANSWER) GO TO 38
IF (JBF .EQ. 4HNEXT) GO TO 38
IF (JBF .EQ. 5HGRADE) GO TO 38
IF (JBF .EQ. 5HPRINT) GO TO 38
IF (JBF .EQ. 6HSTATUS) GO TO 38
IF (JBF .EQ. 4HDONE) GO TO 38
IF (JBF .EQ. 5HXANGE) GO TO 38
IF (JBF .EQ. 4HTEST) GO TO 38
GO TO 35
38 IS=1
IEND=-1
GO TO 450
39 IS=1
IEND=1
JBFN=0
RETURN
40 IF(XS)+1,+3,+3
41 IF(X)+3,+2,+3
42 X=X.E-29
43 X=X*XS
IVAL=X
IS=0
IEND=0
IF (IFLAG-3) +50, ++, +50
44 IS=-1
GO TO 450

```

```
000151      50 XS=-1.0
000153      51 IF IAG=1
000154          GO TO 100
000155      60 IF (IFLAG-3) 65,40,65
000157      65 IFLAG=3
000160          GO TO 100
000161      70 XK=K-IF0
000163          IF (IFLAG-3) 80,90,80
000166      80 X=X*10+XK
000171          IFLAG=2
000172          GO TO 100
000173      90 X=X+XF*XK
000175          XF=XF*.1
000177     100 I=I+1
000200          IF (I-N) 1,1,110
000202     110 IF (IFLAG) 40,39,40
000204     450 IF (K .EQ. ICM) GO TO 492
000206          IF (K .NE. BLANK) RETURN
000210          I=I+1
000211          IF (I .GT. N) RETURN
000214          CALL PULL(I,INLIN,K)
000217          GO TO 450
000223     492 I=I+1
000225          RETURN
000225          END
006400
```


Subroutine FIND

This routine compares two non-numeric fields and returns a percentage agreement coefficient.

Input Parameters

INN -- number of characters in argument 1

IN -- text of argument 1

ITAB -- array containing argument 2. First word is number of characters for argument 2. Next two words are text for argument 2.

Output Parameters

IVAL -- percentage agreement coefficient (0 to 100)

Subroutines Called

CMPAR -- compare two characters

```

000006 SUBROUTINE FIND(INN,IN,ITAB,IVAL)
000006 INTEGER FSUM,ARGWT,AINUX,TBNDX,SUM,DIV,PCOM
000006 DIMENSION IN(1),ITAB(1)
000006 IVAL=0
000006 ARGWT=INN*9
000007 J=ITAB(1)
000010 K=10
000011 DIV=J*9
000013 IF (DIV .LT. ARGWT) DIV=ARGWT
000016 IDIF=INN-J
000017 PCOM=1
000020 I=0
000021 N=0
000021 M=0
000022 SUM=0
000023 100 AINDX=I+N+1
000025 TBNDX=I+K+M+1
000030 CALL CMPAR(AINDX,IN,TBNDX,ITAB,INEQ)
000034 IF (INEQ) 110,110,126
000040 110 PCOM=1
000041 SUM=SUM+9
000043 GO TO 160
000043 126 IF (IDIF) 130, 150, 140
000045 130 IF (M+IDIF) 132, 150, 150
000050 132 M=M+1
000052 GO TO 100
000052 140 IF (N-IDIF) 142, 150, 150
000055 142 N=N+1
000057 GO TO 100
000057 150 IF (PCOM) 220,220,152
000061 152 PCOM=0
000062 160 I=I+1
000064 IF (IDIF) 200, 162, 162
000065 162 IF (J-I) 170,170,100
000070 200 IF (INN-I) 170, 170, 100
000073 220 CALL CMPAR(AINDX-1,IN,TBNDX,ITAB,INEQ)
000101 IF (INEQ .NE. 0) SUM=SUM+3
000106 CALL CMPAR(AINDX,IN,TBNDX-1,ITAB,INEQ)
000114 IF (INEQ .NE. 0) SUM=SUM+3
000121 GO TO 160
000122 170 IVAL=SUM*100/DIV
000126 RETURN
000127 END
007000

```


Subroutine CMPAR

This routine compares two characters.

Input Parameters

AINDX -- position of 1st character

IN -- text buffer containing 1st character

TBNDX -- position of 2nd character

ITAB -- text buffer containing 2nd character

Output Parameter

INEQ -- 0 means equal, 1 means not equal

```
000007 SUBROUTINE CMPAR(AINDEX,IN,TINDEX,ITAB,INEQ)
000007 INTEGER AINDEX,TINDEX,IN(2),ITAB(2)
000011 CALL PULL(TINDEX,ITAB,IT)
000016 CALL PULL(AINDEX,IN,IN1)
000023 IF (IT-IN1) 20,10,20
000024 10 INEQ=0
000024 RETURN
000024 20 INEQ=1
000025 RETURN
000026 END
007400
```


Subroutine PULL

The routine gets a character from an array of text.

Input Parameters

I -- character position

IA -- text buffer

Output Parameter

ICHAR -- right-justified zero-fill character

```
000005 SUBROUTINE PULL(I,IA,ICAR)  
000005 DIMENSION IA(1)  
000010 IWD=(I+9)/10  
000014 ISH=(I+10-IWD*10)*6  
000020 IC=LSHIFT(IA(IWD),ISH)  
000022 ICAR=IC .AND. 77b  
000023 RETURN  
007500 END
```


Subroutine PUSH

This subroutine places a character into an array.

Input Parameters

IBFN -- character position

ICH -- character

Output Parameter

IBF -- array with single character modified

```
000005 SUBROUTINE PUSH(JBFN,JBF,ICH)
000005 DIMENSION JBF(1)
000010 IWD=(JBFN+9)/10
000014 ISH=(JBFN+10-IWD*10)*6
000021 IC=LSHIFT(JBF(IWD),ISH)
000022 IC=(IC .AND. 7777777777777777770000) .OR. ICH
000025 ISH2=60-ISH
000033 JBF(IWD)=LSHIFT(IC,ISH2)
000034 RETURN
007400 END
```


System Routines

Subroutine READPF

This routine obtains a copy of the permanent file specified from the directory whose number is given.

Subroutine SAVEPF

This routine saves the current copy of the file specified in the directory specified (replacing an older version).

Subroutine IOBIN

With the argument specified (70B), this routine "throws away" the current temporary file whose name is given.

Subroutine RENAME

This routine changes the name of the file specified to the new name specified.

Subroutine TIM

This routine obtains the current date and time (hours, minutes, seconds).

Subroutine SECOND

This routine obtains the amount of computer time used by the current job since its initiation in seconds and milliseconds (floating point)

Subroutine RANF

This routine computes a pseudo-random number between 0 and 1 (floating point)

Main Program - UPDATER

This program reads the student records file and allows the professor to display and change the records. The following commands are available:

- 1) Print name. The records of the student named are printed.
- 2) Print all. The records of all students are printed.
- 3) Delete name entry. The entry specified under that student is deleted.
- 4) Change name entry to entry. The entry for that student is changed to the new entry.
- 5) Name entry. The entry is added.
- 6) Quit, stop, or done. The current records are written out and saved. The program is terminated.

Restriction: All records are stored in core. Current dimensions are 50 students maximum, 30 P/R entries maximum for any student.

Subroutines Called

K41 -- get job name
READPF -- read RECRDY file
RDLINE -- read line from terminal
ISCAN -- scanner
SCANR -- get unit, P or R, month, and day
SAVEPF -- save updated RECRDY file

Note: The above routines are the same as the ones in the Tester program with the exception of ISCAN.

68

```

00002 PROGRAM UPDATE(ITTY=77,OUTPUT=ITY,RECMUY)
00002 DIMENSION INAME(50),IUNIT(50),INUM(50),ICODES(30,50),IALF(9,50)
00002 COMMON /ICOM/ IUN,IPR,IMN,IUY
00002 COMMON /INUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JBF(2)
00002 DATA IM/6LRECMUY/
00002 CALL HAL(4LGSNP,JOBNAM)
00004 IF ((JOBNAM .AND. 37777777777000000000) .EQ.
1 03020103637000000000) GO TO 30
00007 PRINT 20
00012 20 FORMAT (*CHAC637 MUST BE USED*)
00012 STOP
00014 30 CALL READPF(4L2+23,IP,IFLAG)
00017 REWIND IP
00021 HEAD (IR,901) NNAMES
00027 901 FORMAT (I5)
00027 DO 100 I=1,NNAMES
00031 READ (IR,902) INAME(I),IUNIT(I),INUM(I), (IALF(J,I),J=1,9)
00046 902 FORMAT (1X,A10,2I5,9A10)
00046 JN=INUM(I)
00050 IF (JN .EQ. 0) GO TO 100
00051 READ (IR,903) (ICODES(J,I), J=1,JN)
00061 903 FORMAT (2X,7A10)
00061 J=0
00062 80 J=J+1
00064 IF (J .GT. JN) GO TO 84
00067 IF (ICODES(J,I) .NE. 1H) GO TO 80
00073 JN=JN+1
00073 INUM(I)=JN
00074 IF (J .GT. JN) GO TO 84
00080 DO 82 K=J,JN
00106 82 ICODES(K,I)=ICODES(K+1,I)
00110 GO TO 80
00111 84 IF (JN .EQ. 1) GO TO 100
00113 JN,I=JN-1
00114 GO TO 60
00117 66 IT1=ICODES(J,I)
00120 ICODES(J,I)=ICODES(J+1,I)
00122 ICODES(J+1,I)=IT1
00123 60 DO 70 J=1,JNM1
00125 CALL PULL(I,INLIN,K)
00127 DECODE (10,62,ICODES(J,I)) IT11,IT12,IT13,IT14
00147 62 FORMAT (I2,1X,K1,13,1X,I2)
00147 DECODE (10,62,ICODES(J+1,I)) IT21,IT22,IT23,IT24
00167 IF (IT21-IT11) 66, 63, 70
00172 63 IF (IT22-IT12) 70, 64, 66
00175 64 IF (IT23-IT13) 66, 65, 70
00200 65 IF (IT24-IT14) 66, 70, 70
00203 70 CONTINUE
00206 100 CONTINUE
00211 PRINT 104
00214 104 FORMAT (1/* PROGRAM UPDATER*/
1 /* COMMANDS: NAME DATA, PRINT NAME, PRINT ALL, OFLETE NAME DAT
2 /* CHANGE NAME DATA TO DATA*/* READY*)
00214 110 CALL HDLINE
00215 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
00224 IF (IEND .EQ. 1) GO TO 110
00226 IF (IEND .NE. -1) GO TO 992
00230 IF (JBF .EQ. 4HQUIT) GO TO 200

```

BEST COPY AVAILABLE

```

00230      IF (JBF .EQ. 4HSTOP) GO TO 200
00232      IF (JBF .EQ. 4HDONE) GO TO 200
00234      IF (JBF .EQ. 5HPRINT) GO TO 300
00236      IF (JBF .EQ. 6HDELETE) GO TO 400
00240      IF (JBF .EQ. 6HCHANGE) GO TO 500
00242      DO 120 I=1,NNAMES
00247      120 IF (INAME(I) .EQ. JBF) GO TO 130
00252      122 PRINT 990
00256      990 FORMAT (* NAME NOT FOUND*)
00256      GO TO 110
00257      130 INM=I
00261      CALL SCANR(ID)
00262      IF (ID .EQ. 3HBAD) GO TO 992
00270      INUM(INM)=INUM(INM)+1
00271      I=INUM(INM)
00272      ICODES(I,INM)=ID
00274      134 IF (IPR .NE. 1HP) GO TO 110
00276      IF (IUN .GT. IUNIT(INM)) IUNIT(INM)=IUN
00302      GO TO 110
00303      992 PRINT 994
00307      994 FORMAT (* DATA ERROR*)
00307      GO TO 110
00310      300 INS1=1
00311      INS2=NNAMES
00313      CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
00321      IF (IEND .NE. -1) GO TO 301
00323      IF (JBF .EQ. 3HALL) GO TO 301
00325      310 DO 312 INS1=1,NNAMES
00333      312 IF (INAME(INS1) .EQ. JBF) GO TO 314
00336      GO TO 122
00337      314 INS2=INS1
00341      301 DO 320 I=INS1,INS2
00343      PRINT 302,INAME(I),IUNIT(I)
00352      302 FORMAT (//2X,A10,* PASSED UNIT=I4)
00352      JN=INUM(I)
00354      IF (JN .EQ. 0) GO TO 320
00355      PRINT 304,(ICODES(J,I),J=1,JN)
00365      304 FORMAT (2X,A10)
00365      320 CONTINUE
00370      IF (INS1 .NE. INS2) GO TO 110
00371      CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
00400      IF (IEND .EQ. -1) GO TO 310
00402      GO TO 110
00403      400 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
00412      IF (IEND .NE. -1) GO TO 992
00414      DO 410 INM=1,NNAMES
00422      410 IF (INAME(INM) .EQ. JBF) GO TO 414
00425      GO TO 122
00426      414 CALL SCANR(ID)
00430      IF (ID .EQ. 3HBAD) GO TO 992
00432      JN=INUM(INM)
00434      DO 420 IUN=1,JN
00444      420 IF (ICODES(IUN,INM) .EQ. ID) GO TO 430
00447      GO TO 992
00450      430 JN=JN-1
00451      INUM(INM)=JN
00452      IF (IUN .GT. JN) GO TO 110
00456      DO 440 I=IUN,JN

```



```

000464 440 ICODES(I,INM)=ICODES(I+1,INM)
000466 GO TO 110
000467 500 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
000470 IF (IEND.NE.-1) GO TO 992
000500 DO 510 INM=1,NAMES
000506 510 IF (INAME(INM).EQ.JBF) GO TO 514
000511 GO TO 122
000512 514 CALL SCANR(ID)
000514 IF (ID.EQ.3HBA0) GO TO 992
000516 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
000525 IF (JBF.NE.2HT0) GO TO 992
000527 CALL SCANR(ID2)
000531 IF (ID2.EQ.3HBA0) GO TO 992
000533 JN=INUM(INM)
000535 DO 520 K=1,JN
000545 520 IF (ICODES(K,INM).EQ.ID) GO TO 530
000550 GO TO 992
000550 530 ICODES(K,INM)=ID2
000554 GO TO 134
000555 200 REWIND IR
000557 WRITE (IR,901) NAMES
000565 DO 210 I=1,NAMES
000567 WRITE (IR,902) INAME(I),IUNIT(I),INUM(I),(IALF(J,I),J=1,9)
000604 JN=INUM(I)
000606 IF (JN.EQ.0) GO TO 210
000607 WRITE (IR,903) (ICODES(J,I),J=1,JN)
000617 210 CONTINUE
000622 REWIND IR
000624 CALL SAVEPF(4L2423,IR,4L3068,IFLAG)
000627 PRINT 220
000633 220 FORMAT (* UPDATE COMPLETE*)
000633 END
004700

```

Subroutine SCANR

This routine calls the ISCAN scanner four times to obtain the unit number, a P or R, the month number, and the date. These items are packed into a single word.

Subroutines Called

ISCAN -- updater version of scanner

PULL -- get a character


```

000002 SUBROUTINE SCANR(ID)
000002 COMMON /ICOM/ IUN,IPR,IMN,IDY
000002 COMMON /INPUTL/ INLIN(8),IBFN,IPT,XVAL,IEND,JBFN,JRF(2)
000002 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JRF)
000011 IF (IEND.NE. 0) GO TO 992
000013 IUN=XVAL
000015 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
000023 IF (IEND.NE. -1) GO TO 992
000026 IPR=JBF
000027 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
000036 IF (IEND.NE. 0) GO TO 992
000040 IMN=XVAL
000042 CALL PULL(IPT,INLIN,ICH)
000044 IF (ICH.NE. 1R/) GO TO 992
000047 IPT=IPT+1
000050 CALL ISCAN(INLIN,IBFN,IPT,XVAL,IEND,JBFN,JBF)
000057 IF (IEND.NE. 0) GO TO 992
000061 IDY=XVAL
000063 ENCODE (10,150,ID) IUN,IPR,IMN,IDY
000100 150 FORMAT (I2,1X,A2,I2,1H/,I2)
000101 RETURN
000102 992 ID=3HBAD
000103 RETURN
000104 END
007200

```

Subroutine ISCAN (Updater version)

This version of the scanner will return alphabetic words as separate words rather than connected. That is, the command DELETE SMITH 3 R 10/25 will be processed as 7 different items instead of a single item.

Subroutines Called

PULL -- get a character

PUSH -- store a character


```

SUBROUTINE ISCAN(INLIN,N,I,X,IEND,JBFN,JBF)
INTEGER BLANK
DIMENSION INLIN(8),JBF(2)
DATA IF0,IF9,IPNT,BLANK,MINUS,IPLUS,IR0,IR9,IR.,IR.,IR-,IR+,
DATA ICA,ICZ,ICM/IMA,IMZ,IR,/
IFLAG=0
X=0.0
XS=1.0
XF=.1
JBFN=0
JBF(1)=1H
IF(I-N)1,1,39
1 CALL PULL(I,INLIN,K)
IF(K-IF0)4,70,9898
9898 IF(IF9-K)4,70,70
4 IF(K-IPNT)5,60,5
5 IF(IFLAG-2)10,40,40
10 IF(K-BLANK)15,100,15
15 IF(IFLAG-1)20,40,40
20 IF(K-MINUS)25,50,25
25 IF(K-IPLUS)30,51,30
30 JBF(2)=1H
34 JBFN=JBFN+1
CALL PUSH(JBFN,JBF,K)
35 I=I+1
IF(I.GT.N) GO TO 38
CALL PULL(I,INLIN,K)
IF(K.EQ.BLANK) GO TO 38
IF(K.EQ.ICM) GO TO 38
IF(JBFN.EQ.20) GO TO 35
GO TO 34
38 IEND=-1
GO TO 450
39 IEND=1
RETURN
40 IF(XS)41,43,43
41 IF(X)43,42,43
42 X=1.E-29
43 X=X*XS
IVAL=X
IEND=0
GO TO 450
50 XS=-1.0
51 IFLAG=1
GO TO 100
60 IF(IFLAG-3)65,40,65
65 IFLAG=3
GO TO 100
70 XK=K-IF0
IF(IFLAG-3)80,90,80
80 X=X*10+XK
IFLAG=2
GO TO 100
90 X=X+XF*XK
XF=XF+.1
100 I=I+1
IF(I-N)1,1,110
110 IF(IFLAG)40,39,40

```

```

000153 450 IF (K .EQ. ICM) GO TO 492
000155 IF (K .NE. BLANK) RETURN
000157 I=I+1
000160 IF (I .GT. N) RETURN
000163 CALL PULL(I,INLIN,K)
000166 GO TO 450
000172 492 I=I+1
000174 RETURN
000174 END
000500

```